

05

OpenCV 사용하기 III

도형 그리기 및 차선 검출

05 Mouse Event

- 학습 목표: 마우스를 클릭하여 커서가 있는 좌표를 확인합니다.
- 메인 코드:

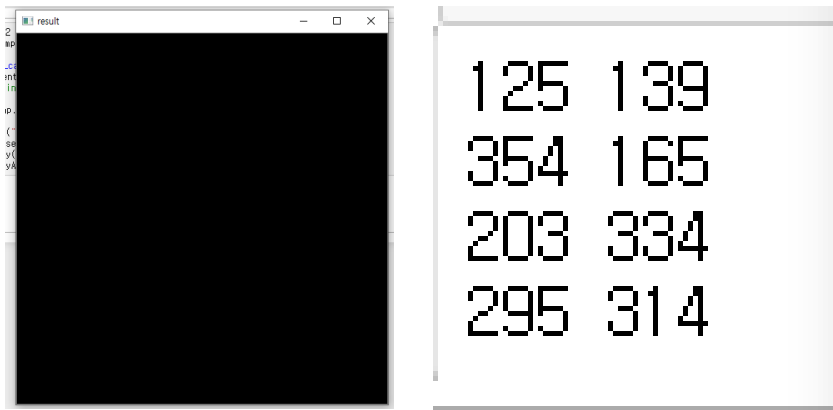
```
In [ ]: import cv2
import numpy as np

def mouse_callback(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        print(x, y)

canvas = np.zeros((500, 500, 3), np.uint8)

cv2.imshow("result", canvas)
cv2.setMouseCallback("result", mouse_callback)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- 결과:



- 코드 설명:

1. `cv2.setMouseCallback`(윈도우창 이름, 콜백함수, 파라미터)

콜백함수에 (event, x, y, flags, param)이 전달되며, 파라미터는 콜백함수에 전달되는 데이터입니다.

05 직선 그리기

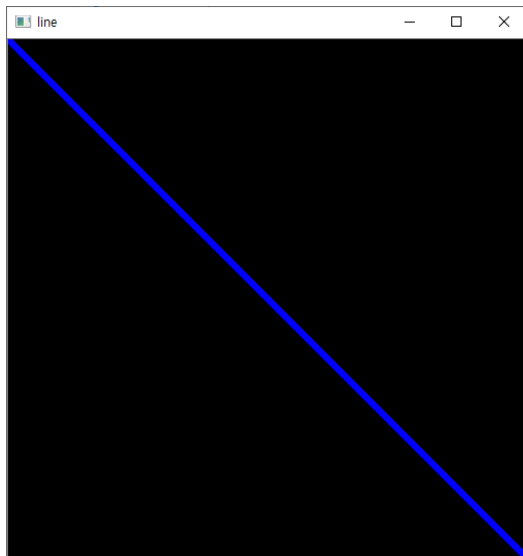
- 학습 목표: 시작점과 끝점을 설정하여 직선 생성하기
- 메인 코드:

```
In [1]: import cv2
import numpy as np

canvas = np.zeros((500, 500, 3), np.uint8)

cv2.line(canvas, (0, 0), (499, 499), (255, 0, 0), 5)
cv2.imshow("line", canvas)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- 결과:



- 코드 설명:

1. `cv2.line` (이미지, 포인트1, 포인트2, 색상, 선두께)

포인트1: 시작점 좌표(x,y), 포인트 2: 종료점 좌표 (x,y)

색상: rgb (0~255), 선두께: 기본 1

05 동그라미 그리기

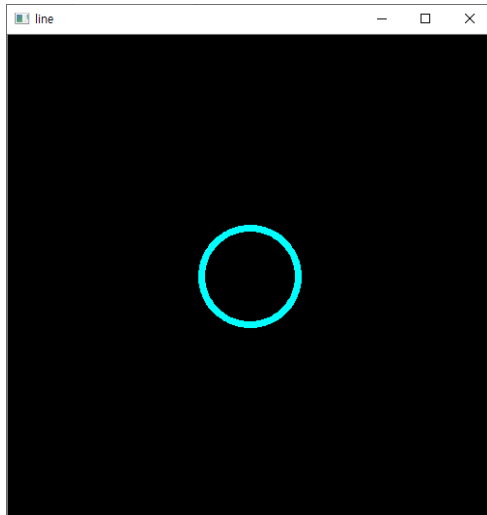
- 학습 목표: 동그라미 생성하기
- 메인 코드:

```
In [1]: import cv2
import numpy as np

canvas = np.zeros((500, 500, 3), np.uint8)

cv2.circle(canvas, (250, 250), 50, (255, 255, 0), 5)
cv2.imshow("line", canvas)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- 결과:



- 코드 설명:

1. `cv2.line` (이미지, 원의 중심, 원의 반지름, 색상, 선두께)

원의 중심: x,y 좌표, 색상: rgb (0~255), 선두께: 기본 1

05 사각형 그리기

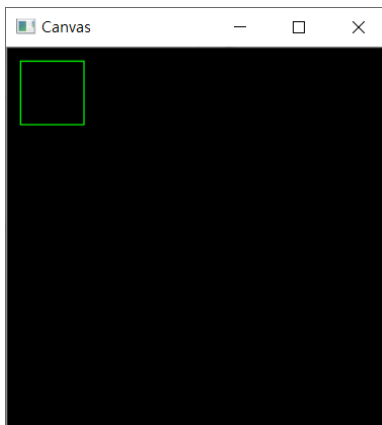
- 학습 목표: 사각형 생성하기
- 메인 코드:

```
import numpy as np
import cv2

green = (0,255,0)
canvas = np.zeros((300, 300, 3), dtype = "uint8")

cv2.rectangle(canvas, (10, 10), (60, 60), green)
cv2.imshow("Canvas", canvas)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- 결과:



- 코드 설명:

1. **cv2.rectangle(canvas, (10, 10), (60, 60), green)**

cv2.rectangle(도형을 그릴 위치, 시작점(사각형 좌측상단), 끝점(사각형 우측하단), 색, 선 크기))

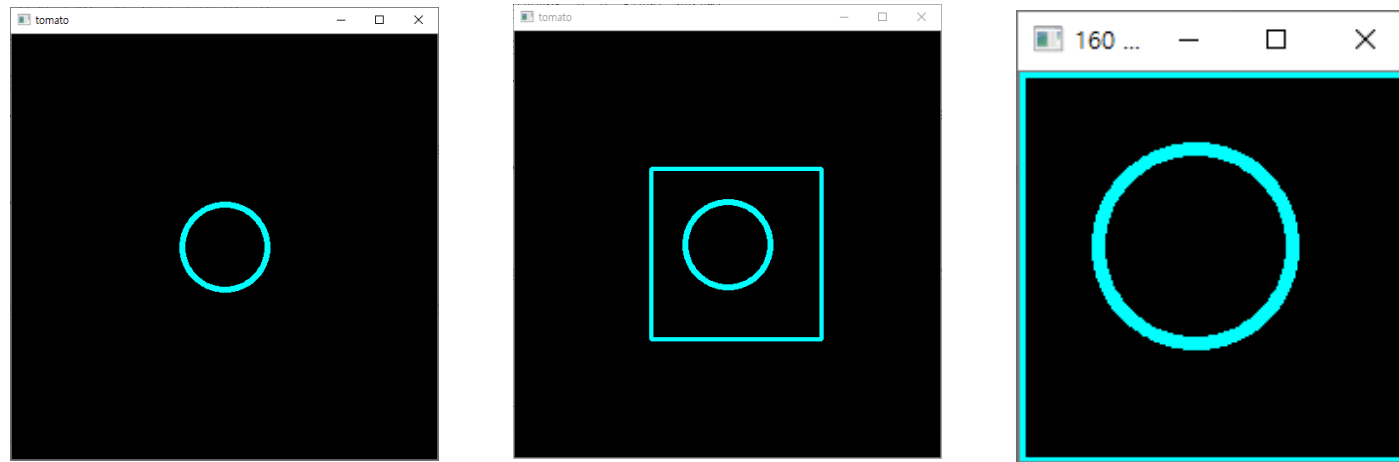
05 문제 4. 마우스 이벤트를 응용하여 ROI 설정하기

- 문제:

1. 배경이 검정색이 화면에 동그라미를 생성합니다.
2. 한 개의 포인트를 찍고 그 포인트로부터 $X+200$, $Y+200$ 인 사각형을 생성합니다.
3. 생성된 사각형을 잘라내어 새로운 윈도우 창에서 표시합니다.

*Note: 마우스 이벤트 사용하기

- 결과:



05 문제 4. 차선 인식하기

- 학습 목표: 흰색과 노란색 차선을 인식합니다.
- 메인 코드:

```
In [*]: import cv2  
  
img = cv2.imread("Image/road.jpg")  
  
hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)  
  
mask = cv2.inRange(hsv, (0, 0, 140), (119, 55, 255))  
mask = cv2.bitwise_and(hsv, hsv, mask=mask)  
result = cv2.cvtColor(mask, cv2.COLOR_HSV2BGR)  
  
cv2.imshow("original", img)  
cv2.imshow("result", result)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

- 결과:



- 코드 설명:

1. `cv2.inRange`(단일 채널, 최소값, 최대값)

단일채널: Hue(색상)의 범위를 조절하여 특정 색상만 표시합니다.

05 문제 5. 차선 검출하기

- 문제:

도로 이외 영역을 없애고 검출된 도로의 엣지를 표시합니다.

(ROI처리, 허용사항 = ROI범위 안에 차량 혹은 장애물은 정답처리)

- 도로에 2가지 색상이 존재할 시 2개다 검출 가능해야한다.
- 검출된 도로는 엣지 검출을 할 것.
- 차선 이외(허용사항 제외) 엣지는 존재해서는 안된다.

- 결과:

